

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Janez Štular

Sistem za upravljanje z recepti

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Uroš Lotrič

Ljubljana, 2016

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Proizvodna podjetja morajo v nenehnem boju za kupce stremeti k čim bolj prilagodljivi proizvodnji s kar najmanj napakami. Pri tem jim je v veliko pomoč programska oprema, ki določene postopke avtomatizira. Razvijte sistem za avtomatsko upravljanje z recepti, ki vključuje spletno aplikacijo za izdelavo receptov in njihovo nalaganje na proizvodne stroje. Programska oprema naj zagotavlja integriteto podatkov in naj bo izdelana po priporočilih GAMP.

Zahvaljujem se mentorju izr. prof. dr. Urošu Lotriču za vso strokovno pomoč in nasvete pri izdelavi diplomske naloge. Za vso podporo se zahvaljujem tudi družini, prijateljem in vsem sodelavcem.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Arhitektura sistema	3
2.1	Spletna aplikacija za upravljanje z recepti	5
2.2	Komunikacijski računalnik	5
2.3	Stroji s strežnikom OPC DA	6
3	Podatkovni tok	9
4	Postopek nalaganja receptov	15
4.1	Vmesnik Kepware in odjemalec OPC DA	26
4.2	Podatkovna baza Microsoft SQL	28
5	Spletna aplikacija za upravljanje z recepti	31
5.1	Proizvodnja	33
5.2	Poročila	34
5.3	Recepti	36
5.4	Šifranti	38
5.5	Zaščita	39

6	Projektna dokumentacija po priporočilih GAMP5	41
6.1	Uporabniške zahteve	41
6.2	Funkcijska specifikacija	42
6.3	Specifikacija programske opreme	43
6.4	Specifikacija strojne opreme	43
6.5	Specifikacija konfiguracije	44
6.6	Testiranje	44
6.7	Uporabniška navodila	47
6.8	Način obvladovanja dokumentov	48
7	Sklepne ugotovitve	51
	Literatura	53

Seznam uporabljenih kratic

kratica	angleško	slovensko
OPC	Object Linking and Embedding for Process Control	Povezovanje in vgradnja objektov za procesni nadzor
OPC UA	OPC Unified Architecture	poenotena arhitektura OPC
OPC DA	OPC Data access	podatkovni dostop OPC
COM	Component Object Model	Komponenta objekt model
DCOM	Distributed COM	Porazdeljeni COM
SQL	Structured Query Language	Strukturiran povpraševalni jezik
ODBC	Open Database Connectivity	Prosta povezava na podatkovne baze
IIS	Open Database Connectivity	Internetni informacijski servisi
FTP	File Transfer Protocol	Protokol za prenos datotek
GAMP	Good Automated Manufacturing Practice	Dobra proizvodnja praksa avtomatizacije proizvodnje
URS	User Requirements specification	Specifikacija uporabniških zahtev
FS	Functional Speceification	Funkcijska specifikacija
CS	Configuration Specification	Specifikacija konfiguracije
SDS	Software Design Speceification	Specifikacija programske opreme
HDS	Hardware Design Speceification	Specifikacija strojne opreme
IQ	Instalation Qualification	Kvalifikacija namestitve
OQ	Operational Qualification	Kvalifikacija delovanja

Povzetek

V diplomski nalogi je opisan sistem za upravljanje s proizvodnimi recepti. Vsak izmed njih mora vsebovati ustrezne parametre, ker z njimi opišemo produkt, ki ga proizvaja stroj.

Zato morajo biti recepti ustrezno kreirani, potrjeni in podpisani. Podpisane recepte je potrebno naložiti na stroj, po posebnem komunikacijskem protokolu. Potrebno je vse od preverjanja povezave do prenašanja, preverjanja in shranjevanja parametrov recepta.

Posamezne akcije aplikacije so zaščitene z različnimi dostopnimi nivoji uporabniških skupin. Vse spremembe podatkov se beležijo v zgodovini dogodkov, kar nam zagotavlja integriteto podatkov.

Opisana je tudi celotna arhitektura, struktura spletne aplikacije in dobra proizvodnja dokumentacijska praksa GAMP5.

Ključne besede: procesno vodenje, podatkovna baza SQL, protokol OPC, spletna aplikacija, recepti, GAMP5.

Abstract

The thesis describes the system used for managing production recipes. Each of them must contain the appropriate parameters which describe a product that a machine produces. Therefore, all recipes must be properly created, confirmed and signed.

The signed recipes must then be transferred to a machine, which requires an appropriate communication protocol. Validation is required all the way from the connections, checkups and saving of the recipe's parameters.

The individual application actions are protected with several different user group access layers. All changes made to data are logged into a chronologically organized event log, which ensures the integrity of data.

Finally, the thesis also describes the architecture, the structure of the web application and an example of good automated manufacturing practice GAMP5.

Keywords: process control, SQL database, OPC protocol, web application, recipes, GAMP5.

Poglavje 1

Uvod

V zadnjem času se v industriji pojavlja potreba po enotnemu vodenju in upravljanju proizvodnih procesov. Podjetja imajo željo po integraciji celotnega sistema vodenja v enoten sistem, preko katerega lahko manipulirajo s podatki o proizvodni in jih nalagajo na same stroje. Potrebe po tem se pojavljajo zaradi vse večjega števila strojev in še večjega števila njihovih proizvodov, pri čemer lahko več strojev proizvaja enak proizvod.

Recepti se vnašajo na vsak stroj posebej. Vnos parametrov recepta poteka preko pripadajočih terminalov, ki ne vsebuje klasične tipkovnice in miške. Stroji s terminali se nahajajo v zelo različnih obratih vse od eksplozijsko nevarnih do čistih in sterilnih. Zaradi ročnega razmnoževanja proizvodnih receptov prihaja tudi do tipkarskih napak.

Problem nalaganja receptov na stroje je rešljiv na več načinov. Za vsak tip stroja lahko naredimo svoj gonilnik in ga vključimo v spletno aplikacijo, kar pa ni primerno zaradi prevelikega števila strojev. V tem primeru, bi morali za vsak nov tip stroja spisati svoj gonilnik, ki ga je potrebno tudi vzdrževati. Takšna rešitev bi izkoriščala strojne zmožnosti komunikacije, a bi bila preveč dovzetna na napake, do katerih bi prišlo pri razvijanju posameznih gonilnikov.

S stališča razvijalcev in vzdrževalcev sistema je bolj primeren enoten način komunikacije za vse stroje, ki pa se razlikuje le v podatkih ki jih prenašamo.

Tako je bila prva ideja uporabiti protokol OPC UA, ki pa žal danes v industriji še ni dovolj razširjen. Bistveno bolj razširjen je njegov predhodnik, protokol OPC DA. Kar je njegova dobra lastnost in je bila odločilna za izbiro načina komunikacije, kljub temu da ima protokol OPC DA veliko varnostnih pomankljivosti in težav pri njegovi konfiguraciji. Z enim načinom komunikacije tako lahko priključimo večje število naprav in nanje nalagamo proizvodnje recepte.

V diplomskem delu sledi opis arhitekture sistema in njegova sestava. Tretje poglavje opisuje podatkovni tok med uporabljenimi strežniki. Glavna tema, opis postopka nalaganja receptov je opisana v četrtem poglavju, kjer so opisani tudi uporabljeni programi. Celoten sistem nalaganja receptov temelji na spletni aplikaciji, ki je opisana v petem poglavju. Celotna aplikacija je dokumentirana po dokumentacijskih priporočilih GAMP5, kar je opisano v šestem poglavju. V zaključku podamo bistvene ugotovitve in dosežene cilje.

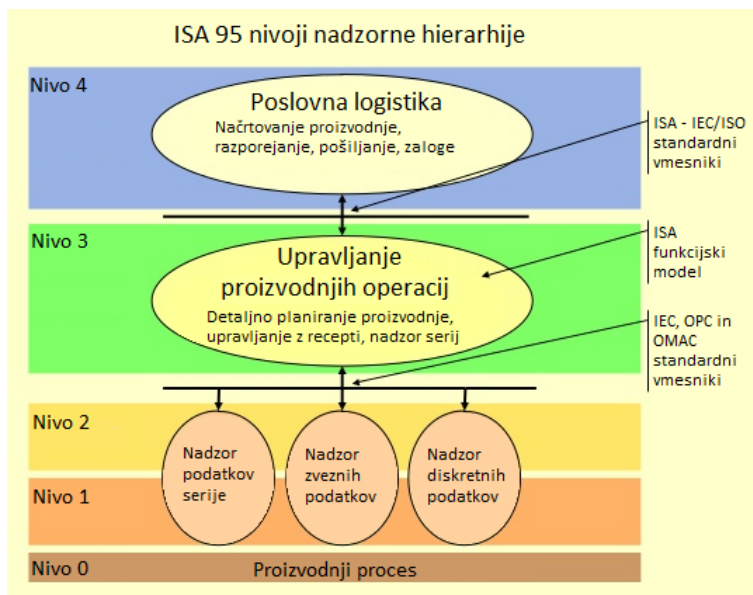
Poglavje 2

Arhitektura sistema

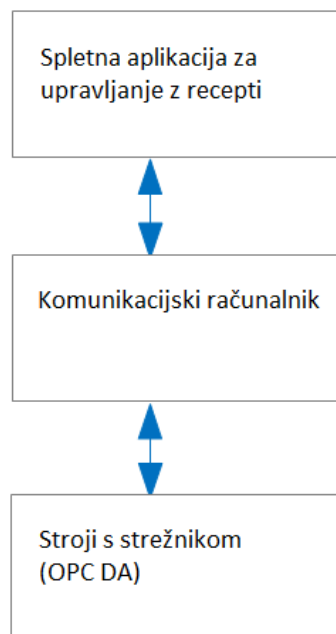
Sistem za upravljanje z recepti je zasnovan po mednarodnem standardu ISA 95, neprofitne organizacije ISA (angl. The International Society of Automation). Ta predpisuje hierarhično zgradbo sistemov avtomatizacije, podatkovne tokove, proizvodne funkcije in podobno, kot je prikazano na sliki 2.1. Sodeč po standardu, naš sistem deluje na dveh nivojih, in sicer na nivoju 2 in na nivoju 3. Spletna aplikacija se nahaja na nivoju 3, stroji s strežnikom OPC DA (angl. Object Linking and Embedding for Process Control Data access) pa na nivoju 2. Nivoja povezuje protokol OPC DA.

Sistem za upravljanje z recepti je namenjen celovitemu upravljanju in nalaganju receptov. Poleg tega lahko iz strojev prenašamo tudi poročila o izvajanju le-teh. Informacijsko gledano je arhitektura sestavljena iz treh medsebojno povezanih gradnikov, kot prikazuje slika 2.2. Ti niso nujno postavljeni na različnih strežnikih, imajo pa povsem različne naloge za delovanje celotnega sistema. Sestavljajo jo:

- spletna aplikacija za upravljanje z recepti,
- komunikacijski računalnik,
- stroji z strežnikom (OPC DA).



Slika 2.1: Hierarhija sistemov vodenja po standardu ISA 95.



Slika 2.2: Arhitektura sistema za upravljanje z recepti.

2.1 Spletna aplikacija za upravljanje z recepti

Spletna aplikacija je namenjena enostavni manipulaciji z recepti. Z njo se na enostaven način izognemo večkratnemu ročnemu vnašanju parametrov na vsak stroj posebej. Poleg tega imamo recepte shranjene na enem mestu, tako da jih lažje vzdržujemo. Spletna aplikacija nam omogoča popolno upravljanje z recepti in je sestavljena iz petih glavnih zavihkov:

- proizvodnja, ki nam omogoča nalaganje receptov,
- poročilo, ki nam prikazuje poročila,
- recepti, ki nam omogoča upravljanje z recepti,
- šifranti, ki nam omogoča urejanje šifrantov,
- zaščita, ki nam prikazuje nivoje zaščite.

2.2 Komunikacijski računalnik

Komunikacijski računalnik je namenjen prenosu podatkov iz enega podatkovnega vira na drug podatkovni vir. Na njem je nameščen program KEPServerEX, ki opravlja nalogo odjemalca in strežnika. Program KEPServerEX med seboj povezuje različne tipe naprav vse od programirljivih logičnih krmilnikov (PLK), strežnikov OPC DA, strežnikov OPC UA in tehtnic do povezave s podatkovnimi bazami. V našem primeru so preko njega povezani stroji s strežniki OPC DA in podatkovna baza Microsoft SQL spletne aplikacije za upravljanje z recepti, kar nam omogoča nalaganje receptov na stroje in preverjanje uspešnosti njihovega nalaganja. Poleg tega služi tudi za prenos poročil, v obliki datotek pdf, iz strojev na spletno aplikacijo za upravljanje z recepti. Na njem je tudi strežnik Proficy Historian A&E, ki zbira alarme in dogodke iz strojev in jih prenaša v podatkovno bazo Microsoft SQL za alarme in dogodke. Komunikacijski računalnik je povezan na dve različni internetni omrežji: na eni strani na proizvodno omrežje, na kateri so stroji,

in na drugi strani na poslovno omrežje, na katerem je strežnik Microsoft SQL s podatkovnimi bazami.

2.3 Stroji s strežnikom OPC DA

Stroji sodijo na nivo procesnega vodenja. Tu se moramo zavedati, da delamo z industrijskimi računalniki in programirljivimi logičnimi krmilniki. Teh ne smemo preobremeniti s programsko opremo, ki bi zahtevala veliko procesorske moči. Prav tako pa moramo biti pazljivi na omejitve prostega pomnilnega prostora. Stroji delajo v realnem času, kar pomeni, da imajo vnaprej postavljene časovne okvirje, v katerih se morajo izvesti njihove operacije, da ne pride do izpadov in podobnih neljubih dogodkov. V našem primeru delamo z industrijskimi računalniki, ki imajo vstavljeno pomnilniško kartico SD z velikostjo 4 GB. Na njej je že nameščen operacijski sistem in strežnik OPC DA. Na nekaterih izmed strojev je naložen operacijski sistem Windows XP, na drugih pa Windows 7. Oba sta seveda v okrnjenih verzijah, primernih za industrijske računalnike. Za komunikacijo z višjim sistemom imamo nameščen strežnik OPC DA. Za uspešno delovanje povezave OPC DA moramo za vsak stroj ustrezno nastaviti DCOM (angl. Distributed Component Object Model) in pravice dostopa do strežnikov OPC DA. Poleg tega moramo za uspešen prenos podatkov na strani stroja dodati identičnega lokalnega uporabnika, kot na komunikacijskem računalniku. To pomeni, da mora imeti uporabnik na obeh napravah enako uporabniško ime in geslo. Le-to pa potrebujemo zaradi vzpostavitve zaupanja med napravami. Da strežnik OPC DA, na stroju vidimo na komunikacijskem računalniku, moramo ustrezno skonfigurirati tudi servis OpcEnum (angl. OPC server enumerator), kateremu nastavimo tudi pravice oddaljenega dostopa DCOM. Ta služi kot nekakšen predstavnik vseh strežnikov OPC DA, ki tečejo na stroju. Na pomnilniško kartico se poleg operacijskega sistema in nameščenih programov shranjujejo tudi poročila o receptih, v obliki datotek pdf, ki jih stroj izvaja. Tako moramo paziti tudi nanje, saj nam dodatno zasedajo prostor, dokler jih ne prečrpamo.

Na stroj smo naložili le tri programe:

- strežnik FTP za prenos poročil v obliki datotek pdf,
- zbiralnik Proficy Historian DA,
- zbiralnik Proficy Historian A&E.

Poglavje 3

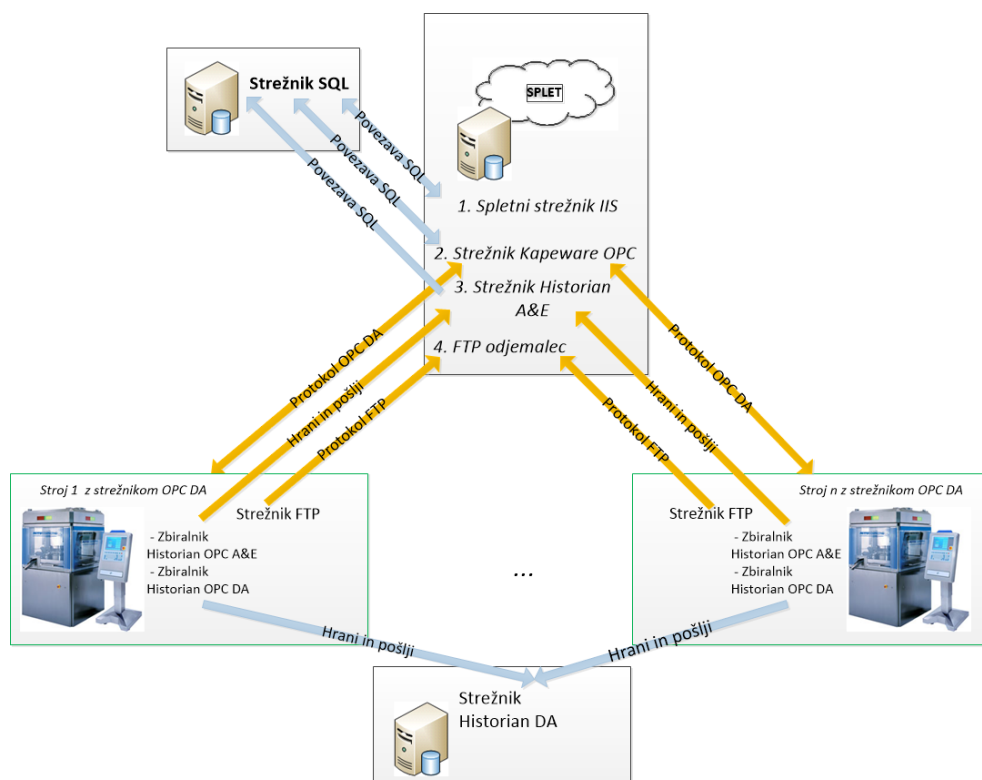
Podatkovni tok

Podatki se v sistemu pretakajo med tremi strežniki in n stroji, pri čemer imajo tudi stroji lahko vlogo strežnika. Arhitektura je prikazana na sliki 3.1 in je sestavljena iz naslednjih naprav:

- strežnika Microsoft SQL,
- strežnika Proficy Historian,
- komunikacijski računalnik in
- poljubnega števila strojev.

Celoten sistem temelji na arhitekturi odjemalec-strežnik, kar nam omogoča enostavno ukinjanje ter dodajanje različnih naprav. Zaradi raznolikosti nameščenih programov, potrebnih za delovanje sistema, pri prenosu podatkov uporabljamo različne tipe povezav:

- povezavo SQL,
- povezavo OPC DA,
- prenos datotek FTP,
- povezavo hrani in pošlji Proficy Historian.



Slika 3.1: Strežniki in njihove prenosne poti.

Za spletni strežnik uporabljamo IIS (angl. Internet Information Server). To je Microsoftov produkt, ki vsebuje strežnike s protokoli:

- spletni strežnik HTTP (angl. Hypertext Transfer Protocol),
- spletni strežnik z varovano povezavo HTTPS (angl. Hypertext Transfer Protocol Secure),
- strežnik za prenos datotek FTP (angl. File Transfer Protocol),
- strežnik za varen prenos datotek FTPS (angl. File Transfer Protocol Secure),
- strežnik za spletno pošto SMTP (angl. Simple Mail Transfer Protocol),
- strežnik novic NNTP (angl. Network News Transfer Protocol).

Na njem teče naša spletna aplikacija za upravljanje z recepti, ki uporablja le spletna strežnika HTTP in HTTPS. Narejena je s tehnologijo ASP.NET Web Forms, pri čemer je uporabljen programski jezik C#.

Delovanje spletne aplikacije temelji na podatkih in konfiguracijah, ki se nahajajo v podatkovni bazi Microsoft SQL, kar je razvidno iz slike 3.1. Za našo aplikacijo uporabljamo centralni strežnik SQL. Za komunikacijo s podatkovno bazo uporabljamo klasičen povezovalni niz SQL, kar pomeni, da se na strežnik Microsoft SQL povezujemo z avtentikacijo SQL in že pri povezovanju podamo ime strežnika, uporabniško ime, geslo in ime vsebinske podatkovne baze.

Strežnik Kepware OPC v našem primeru združuje dva tipa povezav. Povezavo OPC DA, ki povezuje strežnik Kepware OPC z vsakim izmed strojev in povezavo SQL, ki povezuje strežnik Kepware OPC s strežnikom Microsoft SQL. Sestavljata ga dva produkta, ki skupaj tvorita povezavo med proizvodnim in informacijskim nivojem:

- program KEPServerEX, ki je odjemalec OPC DA in odjemalec ODBC (angl. Open Database Connectivity),

- program LinkMaster, ki je namenjen povezovanju (angl. linking) podatkov iz naprav KEPServerEX.

Celoten strežnik Kepware OPC deluje tako, da program KEPServerEX služi kot odjemalec in strežnik, kot je prikazano na sliki 3.1. Tako je na eni strani povezan z vsakim izmed strojev preko povezave OPC DA, na drugi strani pa je odjemalec ODBC, ki je preko povezave SQL povezan na strežnik Microsoft SQL ter njegove tabele SQL. Program KEPServerEX je le strežnik, ki je povezan z različnimi tipi naprav in ne vsebuje nobene logike delovanja. Zato potrebujemo program LinkMaster, ki služi prenašanju podatkov iz naprav OPC DA v tabele SQL in obratno.

Strežnik Proficy Historian A&E je namenjen zbiranju alarmov in dogodkov, ki jih zajema iz zbiralnikov Proficy Historian A&E ter jih zapisuje v podatkovno bazo Microsoft SQL, kot je prikazano na sliki 3.1. Strežnik Proficy Historian A&E za povezovanje na podatkovno bazo ne uporablja avtentikacije SQL, temveč avtentikacijo Windows. To pomeni, da moramo imeti za delovanje povezave na strani strežnika Microsoft SQL in na strani strežnika Proficy Historian A&E kreiranega enakega uporabnika z enakim uporabniškim imenom ter enakim geslom. To mora biti lokalni uporabnik ali pa mora pripadati isti domeni. Za hranjenje alarmov in dogodkov moramo imeti kreirano svojo podatkovno bazo SQL, na kateri mora imeti prej omenjeni uporabnik dovolj pravic. Zbiranje alarmov in dogodkov poteka s pomočjo zbiralnikov Proficy Historian A&E, ki so nameščeni na vsakem izmed strojev. Zbiralniki so navzgor povezani s strežnikom Proficy Historian A&E in navzdol s strojevim strežnikom OPC A&E, ki jim posreduje generirane alarme in dogodke.

Povezave med zbiralniki Proficy Historian A&E, strežnikom Proficy Historian A&E in strežnikom Microsoft SQL imajo funkcionalnost hrani in pošlji (angl. store and forward). To pomeni, da se v primeru padca katerekoli izmed omenjenih povezav podatki hranijo lokalno na stroju ali strežniku Proficy Historian A&E in se pri ponovni vzpostavitvi povezave prenesejo en nivo višje.

Strežnik Proficy Historian DA je namenjen hranjenju procesnih podatkov, ki so v nekaterih družbah zelo pomembni. Tu moramo zelo paziti na integriteto podatkov, saj so napake pri proizvodnji nekaterih izdelkov lahko usodne. Strežnik zbira podatke s pomočjo zbiralnikov Proficy Historian DA, ki so nameščeni na vsakem izmed strojev. Komunikacija med strežnikom Proficy Historian DA in zbiralnikom Proficy Historian DA poteka na način hrani in pošlji (angl. store and forward), ki sem ga razložil v prejšnjem odstavku in kot je prikazano na sliki 3.1.

Odjemalec FTP je bil razvit z namenom, da teče kot opravilo spletne aplikacije. Skonfiguriran je tako, da prenaša poročila, v obliki datotek pdf, izvedenih receptov iz vseh priključenih strojev na mesto spletnega strežnika, kot je prikazano na sliki 3.1. To pomeni, da moramo imeti na vsakem izmed strojev svoj strežnik FTP. Za tako arhitekturo smo se odločili zato, ker so stroji priključeni v proizvodno omrežje in nimajo dostopa do centralnega strežnika Microsoft SQL. Vsako uspešno prenešeno poročilo iz stroja tvori zapis v tabeli SQL. To je pomembno zato, da imamo centralno upravljanje s poročili in jih lahko prikazujemo ter podpisujemo.

Poglavje 4

Postopek nalaganja receptov

Nalaganje receptov izvajamo preko spletne aplikacije. Večina logike potrebne za nalaganje, se nahaja v obliki procedur SQL, ker sta tu najpomembnejša pravilnost parametrov in prepis podatkov. Podatke prenašamo s prepisom vrednosti strojevega strežnika OPC DA v tabele SQL in obratno. Nalagamo lahko le potrjene in aktivirane recepte. Postopek nalaganja je enoličen za vsak tip stroja, z izjemo prvega koraka. V našem primeru ga izvedemo v štirih zaporednih korakih, pri čemer se prva dva koraka izvedeta po kliku na gumb za nalaganje recepta. Zadnja dva koraka se izvajata ob nastavljenih časovnih intervalih in morata upoštevati status nalaganja recepta. Ti koraki so:

- preverjanje povezave,
- preverjanje stroja in statusa recepta,
- preverjanje recepta,
- shranjevanje recepta.

Sledi podrobnejši opis nalaganja receptov, ki je zaradi večje preglednosti predstavljen v obliki diagramov poteka.

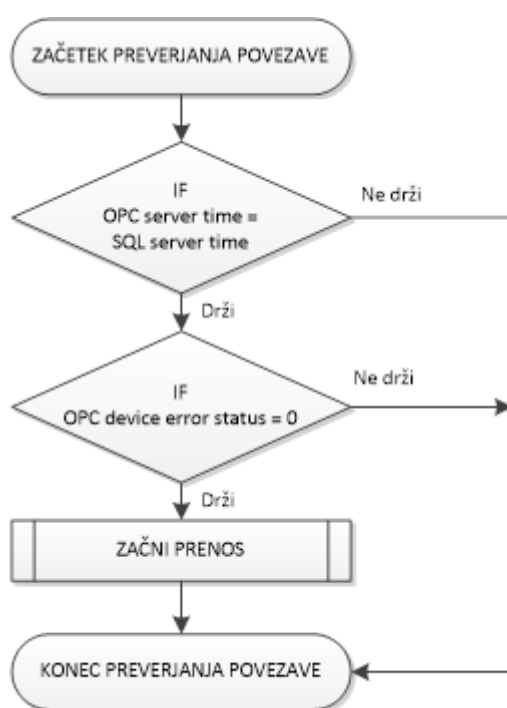
Preverjanje povezave je prvi korak pri nalaganju receptov. Je tudi edini korak, ki je enak za vse naprave, ker ni odvisen od njihovega tipa, ampak

le od komunikacijskega strežnika OPC. Poleg začetka in konca vsebuje dve glavni preverjanji, prikazani na sliki 4.1.

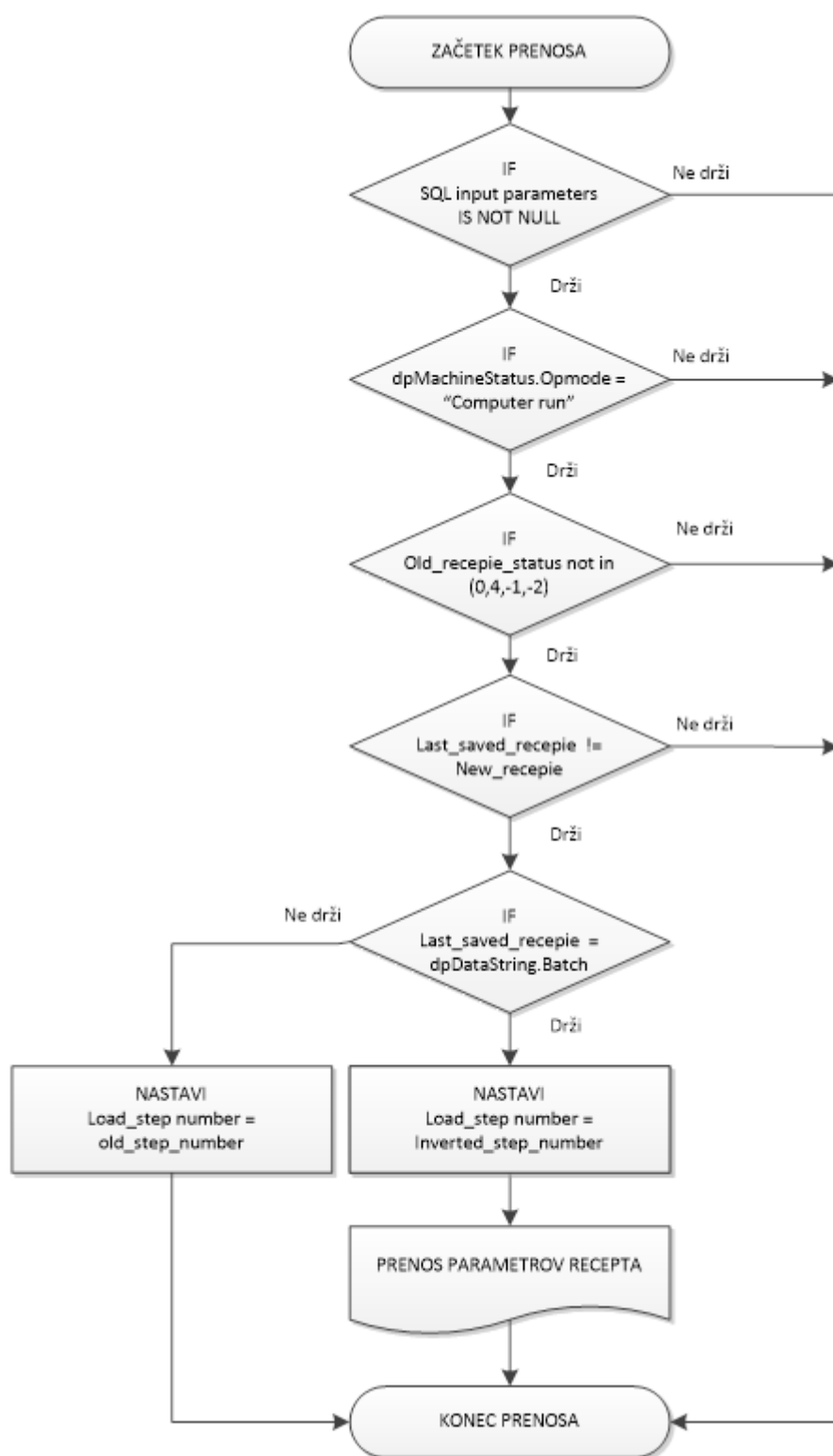
1. Preverimo čas na strežniku OPC in strežniku SQL. Ker je aplikacija zasnovana za globalno uporabo, za primerjavo uporabljamo čas UTC (angl. Coordinated Universal Time). Zato je pogled nanj smiseln iz vsakega dela sveta, neodvisno od tega, v katerem časovnem pasu se nahajamo. Prav tako ne preverjamo točne enakosti časov, temveč časovno razliko, ki mora biti manjša od nastavljene. V primeru, da imata časa preveliko odstopanje, vemo, da je s komunikacijskim strežnikom OPC nekaj narobe.
2. Preverimo status stroja, na katerega želimo naložiti recept. To naredimo tako, da preberemo polje stroja, ki nam prikazuje vrednost napake za povezavo. Če napake na napravi ni, je vrednost tega polja 0, v nasprotnem primeru pa 1 ali -1.
3. Po uspešno končanem preverjanju povezave sledi klic funkcije za nalaganje recepta. V nasprotnem primeru pa moramo nalaganje prekiniti. Poleg tega moramo v primeru napake, le-to prikazati v spletni aplikaciji. Prikažemo lahko dve različni napaki, odvisno na katerem delu preverjanja povezave pride do nje.

Po uspešno zaključenem prvem koraku se nalaganje nadaljuje s preverjanjem stroja in statusa recepta, kar prikazuje slika 4.2. To preverjanje je zelo pomembno, ker stroji niso vedno pripravljeni na nalaganje recepta, poleg tega pa se za vsak recept, ki ga nalagamo, vodi statusno polje.

1. Najprej preverimo vhodne parametre spletne aplikacije in se s tem izognemo anomalijam, do katerih lahko pride zaradi manjkajočih parametrov. Do tega lahko pride, ker je aplikacija pripravljena za nalaganje receptov na različne tipe strojev.



Slika 4.1: Postopek preverjanja povezave.



Slika 4.2: Postopek nalaganja recepta in preverjanje njegovih stanj.

2. Nato preverimo trenutni način delovanja stroja. V našem primeru imamo naslednje načine delovanja stroja:

- neznan (angl. unknown),
- delujoči (angl. computer run),
- testni (angl. test run),
- nastavitveni (angl. setting run)
- standardni (angl. standard run).

Nalaganje recepta je mogoče samo v delujočem načuni.

3. Preveriti je potrebno, v kakšnem stanju je bil recept, ki smo ga nazadnje naložili na stroj. To je pomembno, ker lahko uporabnik med procesom nalaganja recepta ponovno klikne na gumb za nalaganje recepta in s tem ponovno sproži celoten postopek. Do takšnega primera lahko pride, ker nalaganje poteka iz spletne aplikacije in ima lahko tudi občasne zakasnitve in prekinitve. Poleg tega tudi strežniki OPC, ki se nahajajo na strojih, niso tako zelo odzivni. Prav tako se morajo njihovi podatki prepisati tudi v podatkovno bazo Microsoft SQL.

4. V nadaljevanju zaradi uporabniških anomalij preverimo tudi oznako recepta. Na stroju ne potrebujemo dveh identičnih receptov, saj bi s tem le zapravljali pomnilniški prostor in povečali možnosti za napake v primeru naknadnih popravkov parametrov. Do teh lahko pride, ko tehnolog naknadno ponastavlja parametre zaradi optimalnejšega delovanja stroja.

5. Nato moramo preveriti, kateri izmed receptov na stroju je trenutno aktiven. Pomembno je, da ga ne prepíšemo, ker bi v tem primeru dosegli nepotrebno napako. Poleg tega bi lahko z nesrečnim spletom okoliščin recept mogoče še celo prepisali.

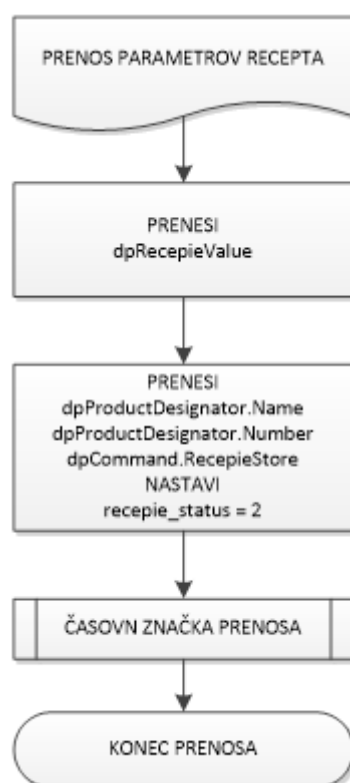
- V primeru, da se na pomnilniškem mestu, kamor želimo zapisati nov recept, nahaja trenutno aktiven recept, moramo pomnilniško

mesto zamenjati. Recepte shranjujemo na prvi dve pomnilniški mesti, in sicer na mesto 1 in na mesto 2. Menjava pomnilniškega mesta je preprosta; v primeru da je bil prejšnji recept zapisan na pomnilniško mesto 1, novega zapišemo na pomnilniško mesto 2 in obratno.

- Če pa zadnji naloženi recept trenutno ni aktiven, ga lahko mirno prepíšemo.
6. Po nastavljenem pomnilniškem mestu za zapis recepta lahko pričnemo z nalaganjem njegovih parametrov na sam stroj. To naredimo v naslednjem koraku, ki mora biti varovan s transakcijo, da ne izgubimo kakšnega podatka. To pomeni, da moramo uspešno prepisati vse parametre. V primeru, slabega prepisa vsaj enega parametra moramo vrednosti vseh parametrov povrniti v prejšnje stanje.

Sledi nalaganje parametrov recepta na samem stroju. Preverili smo že vse pogoje stroja in recepta. Z znanimi parametri stroja lahko pričnemo z nalaganjem parametrov recepta, kot je prikazano na sliki 4.3.

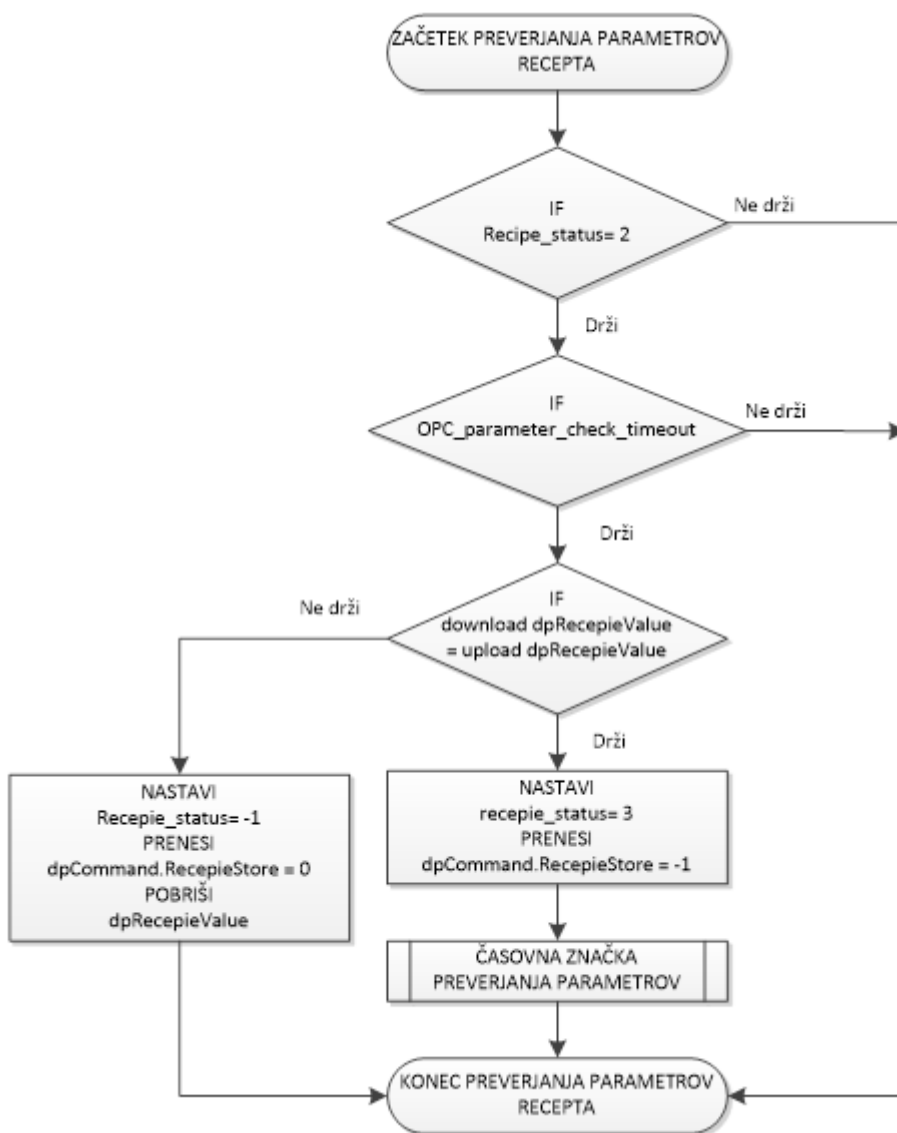
1. Najprej pričnemo z nalaganjem tehničnih parametrov recepta, ki so pomembni za delovanje stroja. Teh je največ in poleg tega morajo biti zapisani prvi.
2. V nadaljevanju prenesemo opisne parametre za recept, ki ga prenašamo. Prav tako prenesemo tudi prej pripravljeno zaporedno številko pomnilniškega mesta, kamor shranimo recept ter nastavitve vrednosti polja, ki aktivira nalaganje recepta na sam stroj. Poleg vsega naštetega nastavimo njegov status na vrednost 2. Ta nam pove, da so parametri recepta uspešno prenešeni.
3. Dodamo mu časovno značko, da bomo v naslednjem koraku vedeli, koliko časa je preteklo od trenutka, ko je bil recept uspešno prenešen na stroj.



Slika 4.3: Postopek nalaganja recepta in njegovih parametrov.

Naslednji korak je preverjanje parametrov recepta na samem stroju. Pri prenosu parametrov moramo biti prepričani, da so se pravilno prenesle vse vrednosti brez izjeme. V nasprotnem primeru moramo pobrisati parametre, postopek je prikazan na sliki 4.4.

1. Postopek preverjanja parametrov receptov se zaporedno izvaja za vse recepte v fazi nalaganja in je enak za vse stroje istega tipa. Zato moramo pred preverjanjem samega recepta preveriti njegov status, ki mora imeti vrednost 2.
2. Zaradi razloga, omenjenega v prejšnji točki, za vsak recept hranimo tudi časovno značko. Parametre recepta začnemo primerjati šele po vnaprej nastavljenem preteku časa. Če čas za nalaganje našega recepta še ni potekel, le-to prekinemo. Ob naslednji iteraciji preverjanj bomo ponovno pričeli s primerjanjem parametrov recepta. Do tega pride zaradi zakasnitev na strežniku OPC in morebitnih motenj na omrežju. Tako se izognemo prehitremu preverjanju in povečamo stopnjo zanesljivosti.
3. Naložene parametre recepta zaradi varnosti preverimo. To naredimo, da parametre, zapisane na stroj, preberemo in jih primerjamo s parametri, ki smo jih želeli zapisati. Po primerjavi vseh parametrov vidimo, ali so bili uspešno prenešeni.
 - Če so parametri uspešno prenešeni, so vrednosti naloženih parametrov enake vrednostim prebranih parametrov. V tem primeru dobi recept status 3 in je pripravljen na naslednji korak. Poleg tega ne smemo pozabiti nastaviti polja za shranjevanje recepta na vrednost -1. Recept s preverjenimi parametri dobi tudi časovno značko preverjanja, ki pa je pomembna za naslednji korak shranjevanja recepta. Tu pazimo predvsem na odziv strežnika OPC, ki se nahaja na stroju in nam vrača morebitne napake.
 - V primeru, da se vrednost kateregakoli naloženega parametra razlikuje s prebranim parametrom, nalaganje ni uspešno. Receptu

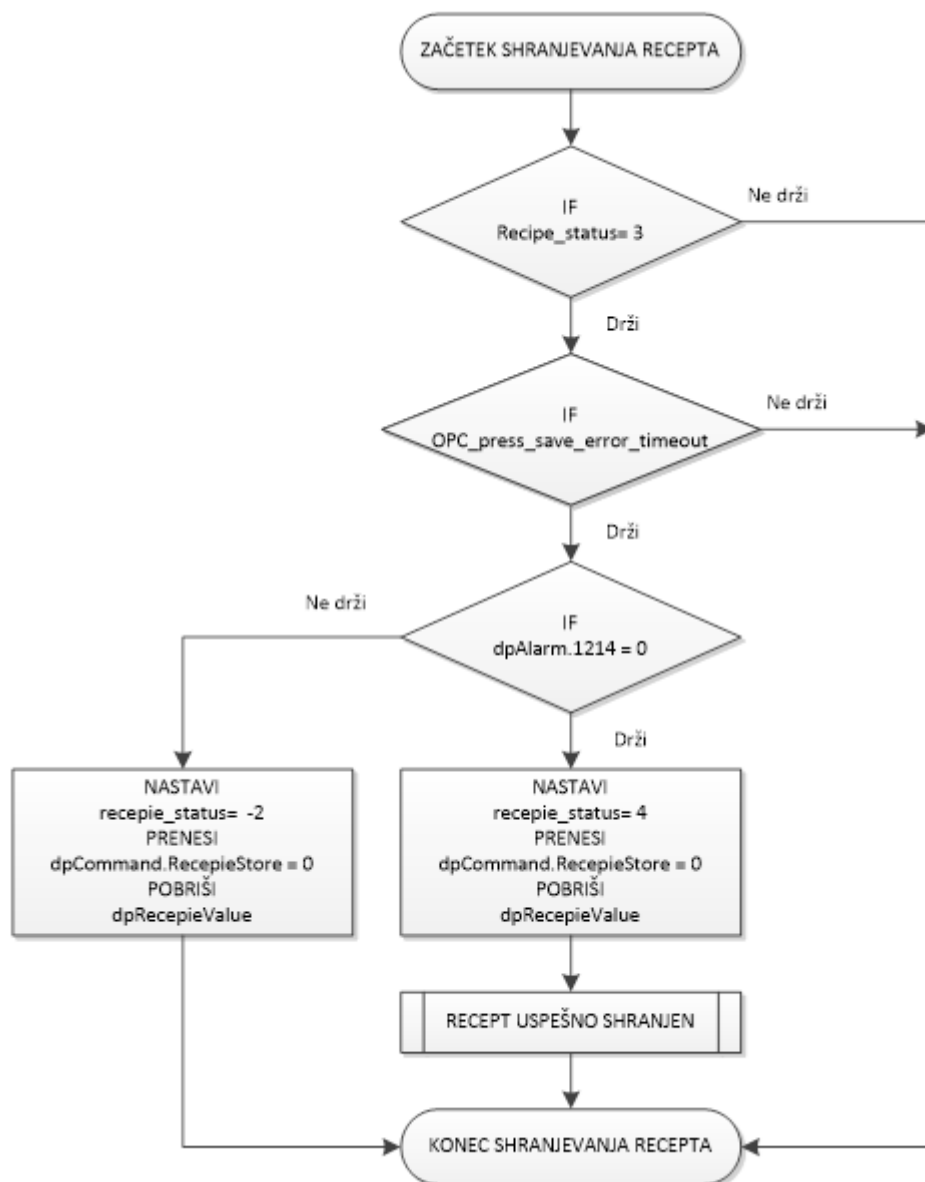


Slika 4.4: Posotopek preverjanja prenosa parametrov recepta.

zato dodelimo status -1, kar pomeni napako pri zapisu parametrov recepta. Poleg tega nastavimo polje za shranjevanje recepta na 0 in pobrišemo tabelo za prenos parametrov recepta.

Uspešno preverjen recept je pripravljen na shranjevanje. To je zadnji korak pri nalaganju receptov, ki je tako kot prejšnji sestavljen iz nekaj pomembnejših preverjanj in prepisov. Pri njegovem izvajanju izvemo, ali smo uspešno prenesli recept. V primeru, da se nam nalaganje poruši v tem koraku, lahko iz naprave točno razberemo, kaj je šlo narobe. Postopek je prikazan na sliki 4.5.

1. Tako kot prejšnji korak, se tudi ta izvaja ob nastavljenih časovnih intervalih. Poleg statusa recepta ponovno preverjamo tudi časovno značko, ki je bila generirana ob koncu preverjanja recepta. Shranjevanje recepta se prične šele po pretečenem nastavljenem času, v nasprotnem primeru moramo počakati na novo iteracijo shranjevanj.
2. Preverimo alarmno polje z oznako 1214. Napaka je opisana v dokumentaciji strežnika OPC DA, ki ga imamo nameščenega na stroju. Zapisano je, da nam oznaka 1214 različna od 0 pomeni napako pri nalaganju recepta.
 - V primeru, da je vrednost napake 0, je bil recept uspešno shranjen. Zato mu status nastavimo na vrednost 4 in ga zabeležimo kot zadnji naloženi recept. Poleg tega si zapomnimo tudi njegovo pomnilniško mesto, 1 ali 2. Polje za shranjevanje recepta postavimo na 0, kar pomeni, da se naknadni prenešeni parametri ne prepisujejo, in nato še pobrišemo tabelo z njihovimi vrednostmi.
 - Če nam je shranjevanje parametrov spodletelo, najprej popravimo status recepta na -2. S tem točno vemo, da je shranjevanje recepta neuspešno zaradi vrednosti enega ali več parametrov. Prav tako ponastavimo polje za shranjevanje recepta in pobrišemo tabelo parametrov. V tem koraku preverjamo tudi polje z opisnim izpisom



Slika 4.5: Postopek shranjevanja recepta.

napake shranjevanja. V spletni aplikaciji se nam točno izpiše, kateri parameter recepta ni bil pravilen, kar pa se zgodi zaradi njihove medsebojne odvisnosti. To se nam zgodi, ker stroj lahko opravlja več nalog. Na primer, če stroj lahko izdeluje okrogle in podolgovate izdelke, moramo pri izdelavi okroglega izdelka podolgovitost nastaviti na vrednost 0. To pa zato, ker stroj parametrov z vrednostjo 0 ne upošteva. Takih in podobnih primerov ni malo in ne moremo vseh predvideti, zato se na tem mestu zanašamo na napako stroja. Po končanem zadnjem koraku je recept uspešno prenešen in pripravljen na izvajanje. Uporabnik lahko sedaj ponovno prične z nalaganjem novega recepta.

Enake prepise v uspešnem in neuspešnem primeru shranjevanja recepta izvajamo le zato, da zmanjšamo število posodabljanj tabel. Varujemo prepisovanje podatkov s transakcijami in zmanjšujemo promet na strežniku Microsoft SQL.

4.1 Vmesnik Kepware in odjemalec OPC DA

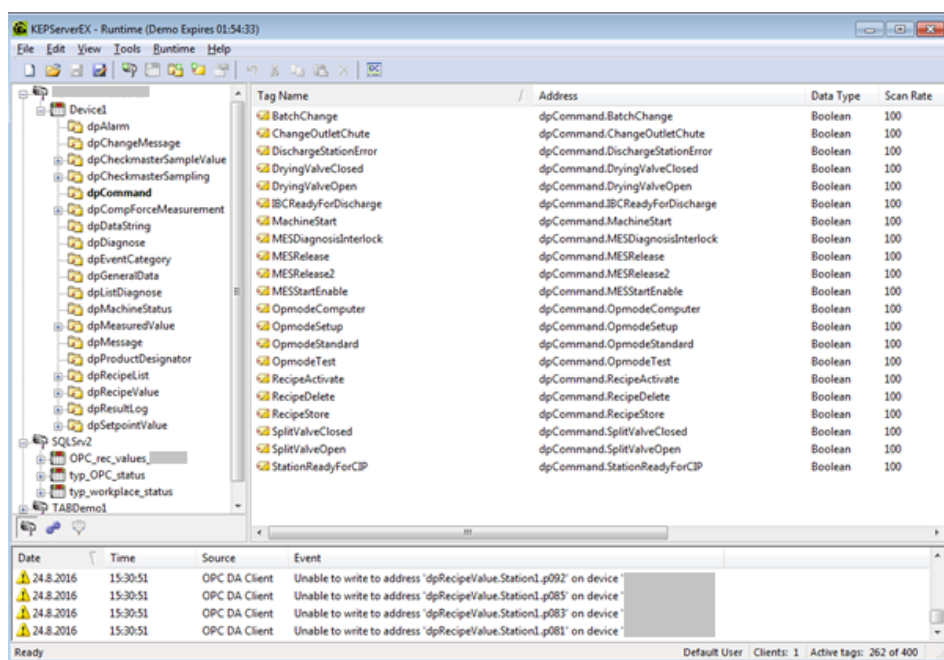
Programski vmesnik Kepware je komunikacijski servis, ki služi prenosu podatkov med različnimi podatkovnimi viri in napravami. V našem primeru je sestavljen iz dveh programov. Prvi ima vlogo komunikacijskega strežnika OPC, drugi pa služi kot pripomoček za prevezavo med napravami, ki se nahajajo na komunikacijskem strežniku. Oba produkta pripadata proizvajalcu Kepware in sta licenčnega tipa. Za potrebe diplomske naloge sta bili uporabljeni preizkusni verziji. Preizkusna verzija nam ponuja polno funkcionalnost produkta, a le za dve uri delovanja. Po pretečenem času moramo strežnika ustaviti in ponovno zagnati.

1. Prvi program je KEPServerEX, ki deluje kot odjemalec in strežnik. Navzven je viden kot strežnik OPC DA in strežnik OPC UA (angl. Unified Architecture). Prav tako je hkrati tudi odjemalec za oba omenjena

protokola. Poleg tega nam program KEPServerEX omogoča povezovanje na 98 % najbolj uporabljenih programirljivih logičnih krmilnikov in ostalih industrijskih naprav, kot so tehtnice. Lahko ga uporabljamo kot pravi komunikacijski strežnik OPC. Poleg omenjenega se z njim lahko povežemo tudi na odjemalca ODBC (angl. Open Database Connectivity) in posledično dostopamo do podatkovne baze. V našem primeru imamo v programu KEPServerEX skonfigurirane tri naprave, kar je razvidno iz slike 4.6, saj imamo tri glavna vozlišča:

- stroj s strežnikom OPC DA, na katerega nalagamo recepte in z njega beremo razna sporočila in napake, v tem primeru KEPServerEX uporabljamo kot odjemalec OPC DA,
- povezavo s strežnikom Microsoft SQL, kjer se nahajajo tabele pomembne za nalaganje recepta, uporabljamo odjemalec ODBC,
- simulacijsko tabelo, ki smo jo zgradili po opisu strežnika OPC DA, ki se nahaja se v proizvajalčevi dokumentaciji in je služila razvoju nalaganja receptov ter povezavi tabel z odjemalcem OPC DA, kot napravo v tem primeru uporabljamo simulator.

2. Drugi program je Kepware LinkMaster, ki ga lahko povežemo na strežnik OPC DA. Program LinkMaster je, podobno kot program KEPServerEX, poleg odjemalca OPC DA tudi strežnik OPC DA. Namenjen je medsebojnemu povezovanju polj različnih naprav. Diagnostika delovanja posamezne prevezave je zelo enostavna. V primeru, da je povezava aktivna in uspešno prepisuje podatke, se obarva zeleno, v nasprotnem primeru pa se obarva sivo. Iz slike 4.7 je lepo razvidno, da v našem primeru deluje le ena povezava, ker v času zajema slike, stroj ni bil priključen. KEPServerEX nam vedno vrača sistemske statuse naprave in če bi v našem primeru preverili status, bi bilo jasno vidno, da je naprava v napaki. Povezave za en stroj so v našem primeru razdeljene v štiri skupine:

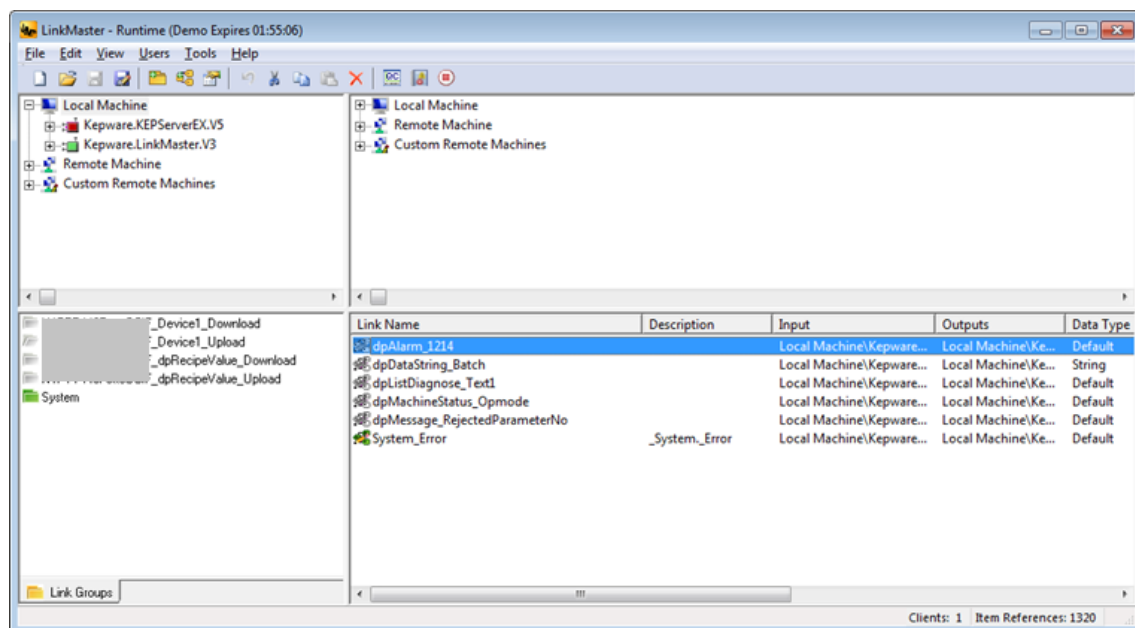


Slika 4.6: Prikaz komunikacijskega strežnika KEPServerEX.

- parametri recepta namenjeni nalaganju na stroj,
- parametri recepta namenjeni branju zapisanih parametrov iz stroja,
- parametri stroja, ki se nanj zapisujejo in služijo nalaganju in shranjevanju recepta,
- parametri stroja, ki se iz stroja berejo in so vidni na sliki. Namenjeni so diagnostiki nalaganja recepta.

4.2 Podatkovna baza Microsoft SQL

Večina logike za nalaganje receptov je napisana v obliki shranjenih procedur SQL. Izsek programske kode Microsoft SQL prvega koraka nalaganja recepta je prikazan na sliki 4.8. Prikazuje pogojna stavka za preverjanje prekinjene povezave, ki oba za status vračata vrednost 1. Ta nam pove, da je prišlo do napake, ki jo prikažemo v spletni aplikaciji. V njej, v primeru, da je vre-



Slika 4.7: Prikaz programa Kepware LinkMaster.

dnost status id postavljena na 1, prikažemo okno z napako pri komunikaciji. V tem oknu se izpiše tudi podrobnejši opis napake v enem izmed spodnjih dveh formatov, odvisno od vzroka, ki je privedel do nje. V primeru, da ne deluje komunikacijski strežnik OPC, sledi opis napake. Sestavljen je iz imena komunikacijskega strežnika in opisa. Ko pa vemo, da komunikacijski strežnik deluje, lahko preverimo tudi delovanje samega stroja. V primeru, ko na komunikacijskem strežniku vidimo, da je postavil sistemsko vrednost za napako stroja na 1, vemo, da imamo napako pri komunikaciji s strojem. V primeru, ko zaznamo katerokoli izmed zgornjih napak, jo prikažemo in zaključimo s preverjanjem povezave. Če uspešno zagotovimo obema pogojevima, lahko pričnemo z naslednjim korakom nalaganja recepta.

```
SELECT @server_time = tos.server_time,  
       @server_name = tos.OPC_server_code,  
       @OPC_code = tw.OPC_code,  
       @device_name = tw.name  
FROM typ OPC_status tos  
     INNER JOIN typ_workplace tw ON tos.OPC_server_code = tw.OPC_server_code  
WHERE tw.id = @typ_workplace_id  
  
IF ABS(DATEDIFF(SECOND,GETUTCDATE(),@server_time)) > (SELECT [dbo].[f_core_setting]('OPC_server_timeout'))  
BEGIN  
    SELECT @typ_workplace_id AS typ_workplace_id,  
           @server_name+' OPC communication server not working' AS status_name,  
           1 AS status_id  
    RETURN  
END  
  
IF (SELECT OPC_device_com FROM typ_workplace_status WHERE OPC_code=@OPC_code) <> 0  
BEGIN  
    SELECT @typ_workplace_id AS typ_workplace_id,  
           @device_name+' OPC Server not working' AS status_name,  
           1 AS status_id  
    RETURN  
END  
  
SELECT @typ_workplace_id AS typ_workplace_id,  
       null AS status_name,  
       null AS status_id
```

Slika 4.8: Izsek programske kode SQL za preverjanje povezave s strežniki OPC.

Poglavje 5

Spletna aplikacija za upravljanje z recepti

Spletna aplikacija za upravljanje z recepti je narejena na podlagi namensko razvitega vmesnika (angl. framework) za grajenje spletnih strani. Tako da za njeno grajenje potrebujemo največkrat le podatke, saj so osnovni gradniki že pripravljeni. Gradnike, ki še niso pripravljeni, lahko v vmesnik dodamo kot elemente po meri (angl. custom part). Nastavitve za prikaz in manipulacijo podatkov na gradnikih pa pridobimo iz podatkovne baze Microsoft SQL. Prav tako so v podatkovni bazi Microsoft SQL shranjene vse ostale konfiguracije spletne aplikacije. Eden izmed osnovnih gradnikov aplikacije je podatkovni vir (angl. data source), pri katerem določimo procedure za izvajanje osnovnih transakcij:

- transakcije za prikaz (angl. select transactions),
- transakcije za vstavljanje (angl. insert transactions),
- transakcije za posodabljanje (angl. update transactions),
- transakcije za brisanje (angl. delete transactions).

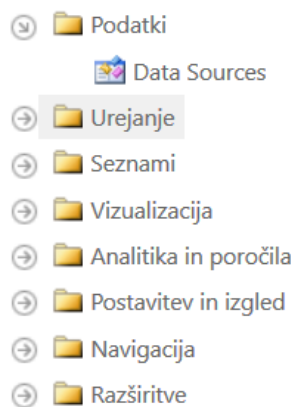
Za pridobivanje podatkov transakcij večinoma uporabljamo procedure SQL (angl. stored procedures) in preko njih izvajamo manipulacijo s podatki.

Ko imamo pripravljen gradnik podatkovni vir, lahko z njegovo pomočjo prevežemo podatke na ostale gradnike.

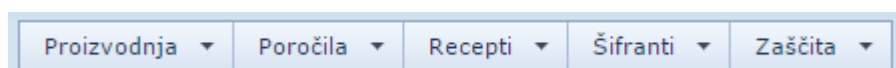
Gradniki za grajenje spletne aplikacije so razdeljeni po skupinah, kot je prikazano na sliki 5.1, in sicer:

- podatki, kamor spada zgoraj omenjena komponenta podatkovni vir;
- urejanje, ki vsebuje različne komponente za urejanje in filtriranje podatkov. Tako sem spadajo predloge za manipulacijo s podatki, kot tudi različne vrstice za parametriziranje in filtriranje podatkov;
- sezname, kamor spada največkrat uporabljen tabelarični način prikaza podatkov (angl. grid), klasičen prikaz podatkov z labelami, polji z besedilom in ostali;
- vizualizacija, ki služi za prikaz grafov, pojavnih oken in vsebovalnikov;
- analitika in poročila, namenjena prikazu in grajenju različnih poročil, vse od procesnih podatkov do sledenja sprememb;
- postavitev in izgled, tu hranimo predloge spletnih strani in stile oblike CSS (angl. Cascading Style Sheets) ter nastavitve izgleda;
- navigacija, kjer imamo različne orodne vrstice in preglede z drevesno strukturo;
- razširitve, kamor spadajo komponente po meri.

Z njimi zgradimo uporabniški vmesnik spletne aplikacije, poleg tega moramo pripraviti podatke v podatkovni bazi Microsoft SQL. Za učinkovito in varno uporabo spletne aplikacije potrebujemo tudi varnostne funkcije. Te so sledenje spremembam, elektronski podpis in ostale funkcionalnosti, ki so dandanes zaradi integritete podatkov zelo pomembne. Dostop do vsebine spletne aplikacije, kot jo vidi uporabnik, poteka preko menijske vrstice, ki jo prikazuje slika 5.2. V njej imamo dostop do vseh petih glavnih zavihkov aplikacije, ki so v nadaljevanju podrobneje opisani.



Slika 5.1: Prikaz osnovnih komponent spletne aplikacije.



Slika 5.2: Prikaz menijske vrstice spletne aplikacije.

5.1 Proizvodnja

Okno proizvodnja je namenjeno nalaganju receptov, prikazano je na sliki 5.3. Sestavlja ga pet glavnih delov, opisanih v nadaljevanju.

1. V hierarhičnem drevesu izberemo stroj, na katerega bomo nalagali recept. V njem izberemo podjetje, lokacijo, obrat in prostor v obratu. Kot listi v drevesni strukturi se nam pojavijo vsi stroji, ki zadoščajo izbranim pogojem.
2. Tabelarni prikaz receptov nam prikazuje recepte, ki jih je mogoče naložiti na izbrani stroj.
3. Gumb nam omogoča pričetek nalaganja izbranega recepta na izbrani stroj. S klikom nanj sprožimo celotni postopek nalaganja recepta.
4. Okno za prikaz statusa nalaganja recepta, ki je sestavljeno iz naslednjih polj:

- statusa naprave, ki nam pove, v kakšnem stanju je naprava in mora biti za nalaganje v stanju (angl. computer run);
- statusa nalaganja recepta, ki prikazuje stanje zadnjega nalaganja recepta ali tekoče stanje trenutnega nalaganja recepta;
- imena recepta, ki smo ga nazadnje naložili ali pa ga trenutno nalagamo;
- zaporedne številke nalaganja, ki nam prikazuje pomnilniško mesto za shranjevanje recepta in ima lahko vrednost 1 ali vrednost 2;
- ukaza za shranjevanje recepta, s čimer vidimo, ali je zapis za shranjevanje recepta aktiven;
- napake, ki nam prikazuje zadnjo zabeleženo napako na stroju, ki je lahko napaka pri delovanju stroja ali pa opis napačnega prenešenega parametra recepta.

5. V statusnem oknu prikazujemo napake pri komunikaciji s strojem. Iz slike 5.3 katerega je lepo razvidno, da je bil izbran stroj v času zajema zaslonske slike v stanju napake. Kar pomeni, da smo imeli povezavo s komunikacijskim strežnikom OPC, s samim strojem pa je ni bilo.

5.2 Poročila

Okno za poročila nam prikazuje dve vrsti poročil. Prva poročila so tista, ki so generirana na stroju ob koncu izvajanja recepta, druga so poročila sledenja sprememb. Prikaz poročil o izvajanju receptov je prikazan na sliki 5.4. Samo okno je sestavljeno podobno kot okno proizvodnja.

1. Hierarhično drevo služi izboru stroja, ki je generiral poročila v obliki datoteke pdf.
2. Filter je namenjen določitvi časovnega intervala sporočil, ki jih želimo prikazati. Poleg tega v filtru lahko nastavimo tudi, katere tipe poročil

Nalaganje receptov

3. NALOŽI PODATKE

5. OPC Server not working

4. Ime recepta: 10-Kopija_16 V.1

Status naprave: Zap. št. nalaganja 1 Ukaz za shranjevanje

Status recepta: -1 Neuspešno shranjevanje

Ime recepta: 10-Kopija_16 V.1

Napaka:

Šifra proizvoda	Naziv proizvoda	Verzija	Tip naprave	Naprava	Status
42	10-Kopija_16	1			Potrjena-aktivna

1.

2.

Slika 5.3: Prikaz okna za nalaganje receptov.

Pregled izvajanja receptov

2. Datum zaklepa: 25.08.2016 00:00 Datum konca: 25.08.2016 23:59

4. Pregled

Repodpisani dokument	Podpisani dokument	Datoteka	Čas prenosa	Serija	Proizvod	Operator	Status	Naprava	Sledenje
		c:\temp-rep-ncv2	25.8.2016 10:13:48		123	REBEV011	Uveljavljen		Sledenje
		c:\temp-rep-ncv2	25.8.2016 10:13:48		123	REBEV011	Pregledan		Sledenje

1.

3.

Slika 5.4: Okno za prikaz in podpisovanje generiranih poročil.

receptov bomo pregledovali. Na voljo imamo dva tipa poročil: Pod tip uvožen spadajo poročila, ki so bila le prenešena preko protokola FTP. Status potrjen pridobijo poročila, ki so elektronsko podpisana.

3. Tabelarični prikaz filtriranih poročil nam omogoča izbor poročila za kasnejše elektronsko podpisovanje. Poleg tega pa lahko natisnemo originalno in elektronsko podpisano poročilo. Obe poročili sta v osnovi isti, le da elektronsko podpisano poročilo vsebuje še podatke o podpisniku in času podpisa.
4. Pritisk na gumb odpre okno za elektronski podpis. V omenjeno okno moramo ponovno vnesti svoje uporabniško ime in geslo, poleg tega moramo ob podpisovanju poročil podati tudi komentar. Ta opisuje, kako se je izvedel recept. Vanj moramo v primeru neuspešno izvedenega recepta napisati, zakaj je do tega prišlo in kaj je bilo pri izvedbi narobe.

Drugo okno za prikaz sledenja sprememb je zelo podobno prvemu, le da ne vsebuje hierarhičnega drevesa in gumba za podpisovanje poročil. Prav tako ima malce drugačno vrstico za filtriranje podatkov, saj le-ta ne vsebuje statusov poročila.

5.3 Recepti

Okno recepti je namenjeno popolni manipulaciji z recepti. V njem lahko kreiramo, urejamo in brišemo recepte. Poleg tega pa lahko urejamo tudi parametre vsakega izmed nepotrjenih receptov. Okno sestavljajo trije glavni deli, kar je razvidno iz slike 5.5.

1. Vrstica za manipulacijo z recepti je sestavljena iz gumbov, katerih uporaba zahteva elektronski podpis. Izjema je le zadnji gumb, ki je namenjen izpisu poročila recepta. Omenjena vrstica je sestavljena iz šestih gumbov.

- Z gumbom Potrdi lahko potrdimo recept, ki ima enega izmed treh statusov. Potrdimo lahko novi neaktivirani recept, novo verzijo recepta ali shranjeno kopijo recepta. Ko je recept enkrat potrjen, ga ni več možno urejati.
 - Z gumbom Aktiviraj lahko aktiviramo recept, aktiviramo lahko le prej potrjene recepte. Aktivacija recepta mora biti podpisana z drugim elektronskim podpisom kot potrditev.
 - Z gumbom Deaktiviraj lahko deaktiviramo recept, deaktiviramo lahko le aktivne recepte.
 - Z gumbom Shrani kot lahko shranimo kopijo izbranega recepta, v kateremkoli stanju. Poimenuje se s prejšnjim imenom, kateremu se pripne niz znakov -kopija. Kopijo recepta je potrebno nato preimenovati in prilagoditi za željeni proizvod. Poleg tega jo je pred nalaganjem potrebno ponovno potrditi in aktivirati.
 - Z gumbom Nova verzija lahko dvignemo verzijo recepta, dvig lahko izvedemo nad zadnjo verzijo izbranega aktivnega recepta. Služi popravljanju in dopolnjevanju recepta za določen proizvod. Z dvigom verzije recepta izgubimo tudi vse podpisnike, tako da je treba novo verzijo ponovno potrditi in aktivirati.
 - Z gumbom Poročilo recepta lahko prikažemo poročilo, z izpisanimi parametri recepta in podpisniki, ki so sprožili zgoraj naštete akcije.
2. Tabelarični prikaz kreiranih receptov nam prikazuje klasične podatke o receptu. Poleg informacij o proizvodu, ki ga izdelujemo, imamo tudi informacije o stroju oziroma napravi ter status recepta. Seveda lahko recepte tudi kreiramo, urejamo ter brišemo.
 3. Podokno za urejanje parametrov nam prikazuje parametre recepta, vezane na izbrano operacijo. To pomeni, da lahko nastavimo vsak parameter, ki pa ima lahko vrednost le med minimalno in maksimalno mejo. Izjema je vrednost 0, ki pomeni, da se parameter recepta ne

Slika 5.5: Prikaz okna za manipulacijo z recepti.

upošteva. Poleg vrednosti ima vsak parameter tudi svoje enote, ki so lahko odstotek, milimeter, mililiter in podobno. Ker so parametri z različnimi enotami, imajo tudi različno stopnjo natančnosti. Tako so lahko nekateri parametri cela števila, drugi pa decimalna. Pri decimalnih številih imamo tudi točno določeno število decimalnih mest. Vse to lahko razberemo iz enote mere ter minimalne in maksimalne vrednosti, ki sta zapisani s točnim številom decimalnih mest.

5.4 Šifranti

Zavihek šifranti je namenjen urejanju šifrantov. Vsebuje vsebinske šifrante sistema, v našem sistemu imamo dva, kar vidimo na sliki 5.6. Eno okno je namenjeno sporočilom, ki se tičejo strojev, drugo pa je namenjeno vozliščem. Pod okno sporočila spadajo alarmi in dogodki zajeti z zbiralnikom Historian A&E, pod okno vozlišča pa spadajo imena teh zbiralnikov. Posamezne šifrante lahko urejamo tako, da jim spremenimo naziv in status, kar je razvidno iz slike 5.6. Vsak šifrant ima lahko status aktivnega, neaktivnega in izbrisanega. Ti statusi so pomembni zaradi prikaza sledenja sprememb, alarmov in dogodkov, ki jih kreirajo stroji.

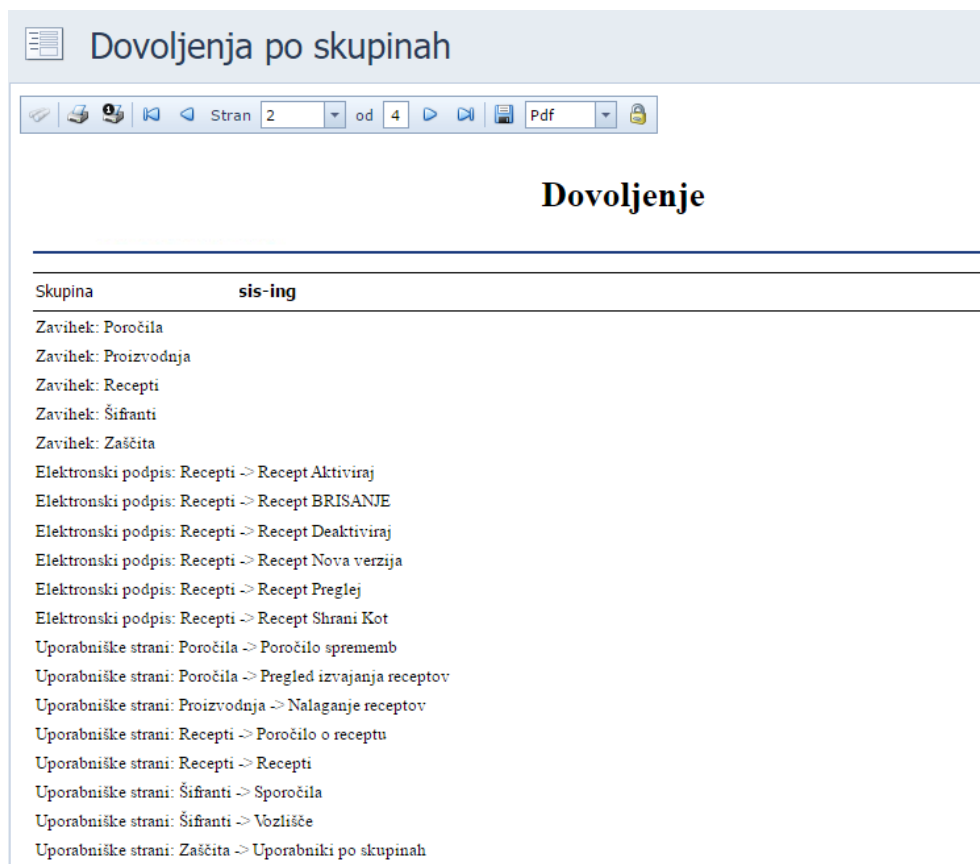
Slika 5.6: Prikaz okna za manipulacijo s šifranti.

5.5 Zaščita

Zavihek zaščita je, podobno kot zavihek šifranti, sestavljen iz dveh oken. Prvo okno je namenjeno prikazu uporabnikov po uporabniških skupinah. Prikazane so le skupine, ki so kreirane. Znotraj vsake izmed skupin pa so naštetih uporabniki, ki spadajo vanje. Na našem sistemu so kreirane naslednje uporabniške skupine:

- administrator,
- sistemski inženir,
- tehnolog,
- tehnik in
- operater.

Uporabniške skupine in pripadnost uporabnikov posamezni skupini je pomembna zaradi omejevanja dostopa do posameznih akcij. To pa nam prikazuje drugo okno. Iz slike 5.7 je razvidno, katere pravice ima skupina uporabnikov imenovana sis-ing, ki je namenjena sistemskim inženirjem. Na sliki



Slika 5.7: Prikaz dovoljenj uporabniške skupine sis-ing.

vidimo, da ni omejen le dostop do pasameznega zavihka, temveč tudi do posameznega okna oziroma uporabniške strani. Poleg omejevanja dostopa do posameznih strani imamo tu določena tudi dovoljenja za elektronsko podpisovanje. Na sliki vidimo, da imajo sistemski inženirji pravico elektronskega podpisovanja le nad oknom recepti, kar pomeni, da lahko generirajo in popravljajo recepte, nimajo pa možnosti elektronskega podpisovanja poročil pdf, ki jih generira stroj ob koncu izvajanja recepta.

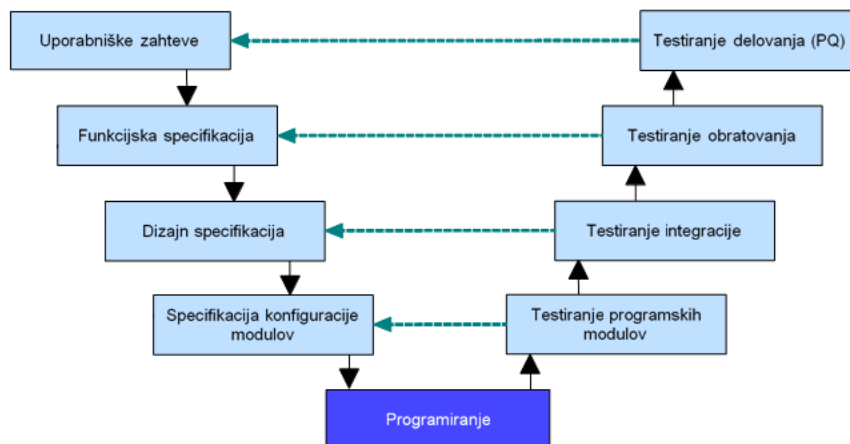
Poglavje 6

Projektna dokumentacija po priporočilih GAMP5

Določeni uporabniki zahtevajo, da se ponudniki opreme držijo priporočil GAMP5 (angl. Good Automated Manufacturing Practice). Ta je sestavljen iz celotnega dokumentacijskega življenjskega cikla, ki predpisuje, katere dokumente potrebujemo in kaj je v njih opisano. Prikazan je na sliki 6.1 in opisuje vse, od vpliva nove aplikacije na obstoječi sistem do samih testiranj. Tako imamo opisane vse kritične elemente, uporabniške zahteve in opis delovanja same aplikacije. Poleg sodijo tudi konfiguracije, uporabniška navodila in testni postopki, ki jih izvedemo nad aplikacijo.

6.1 Uporabniške zahteve

Uporabniške zahteve (angl. User Requirements Specification, URS) je edini dokument, ki ga mora pripraviti naročnik. V njem mora biti jasno opisano, kakšne so želje in zahteve pri izdelavi projekta. Vsem zahtevam, zapisanim v tem dokumentu, moramo kasneje zadostiti in jih tudi primerno testirati. Prav tako je to prvi dokument v dokumentacijskem življenjskem ciklu projekta, ki ga dobi naročnik.



Slika 6.1: Prikaz življenjskega cikla sistema.

6.2 Funkcijska specifikacija

Funkcijska specifikacija (angl. Functional Specification, FS) je dokument, ki opisuje glavno funkcionalnost sistema. To je v našem primeru spletna aplikacija za upravljanje z recepti, ki je v tem dokumentu detajlno opisana. Tako izmed množice poglavij vsebuje dve glavni, zaščito sistema in osnovne funkcionalnosti. V poglavju zaščita sistema je opisano vse, kar se tiče integritete podatkov, to pomeni, da vsebuje teme, kot so:

- zagotavljanje zaščite dostopa, pri čemer omogočimo dostop le uporabnikom določene domene,
- uporabniški dostop in avtentikacija, kjer so opisane zahteve za posameznega uporabnika, kot so uporabniška imena in struktura gesel,
- uporabniške skupine in avtorizacija, ki nam opisuje kreiranje uporabniških skupin, pripadnost uporabnikov in dodelitev pravic,
- potrditev zaščitnih akcij z uporabniškim imenom in geslom ter digitalnim podpisom in
- sledenje spremembam (angl. audit trail).

V poglavju osnovne funkcionalnosti so opisani posamezni zavihki in njihove funkcionalnosti. Zadostiti je potrebno vsem točkam iz uporabniških zahtev. Poleg tega mora biti tudi jasno razloženo delovanje posameznih funkcionalnosti.

6.3 Specifikacija programske opreme

Specifikacija programske opreme (angl. Software Design Specification, SDS) nam opisuje vso programsko opremo sistema, vse od operacijskega sistema do nameščenih programov. Poleg tega imamo tu zapisane tudi njihove osnovne nastavitve. V našem primeru imamo v dokumentu popisane nastavitve, kot so:

- verzije in namestitvene poti posameznih programov,
- nastavitve operacijskega sistema,
- kvaliteto in nastavitve zaslonov,
- nastavitve požarnega zidu in
- naslovi naprav IP (angl. Internet Protocol) in strojni naslovi naprav MAC (angl. Media Access Control).

6.4 Specifikacija strojne opreme

Specifikacija strojne opreme (angl. Hardware Design Specification, HDS) opisuje posamezne dele strojne opreme. V našem primeru imamo le računalnike oziroma strežnike, zato ta dokument ni obširen. Zato vsebuje podatke, kot so:

- količina systemskega pomnilnika,
- velikost trdega diska,

- število procesorjev, njihova moč in število jeder,
- število mrežnih kartic.

Dokument je obsežnejši v primeru avtomatizacijskih projektov, saj so tu opisani vsi programirljivi logični krmilniki, njihovi moduli, regulatorji in ostali pripadajoči elementi.

6.5 Specifikacija konfiguracije

V specifikaciji konfiguracije (angl. Configuration Specification, CS) moramo imeti opisano konfiguracijo sistema. Tako imamo v našem primeru tu opisane:

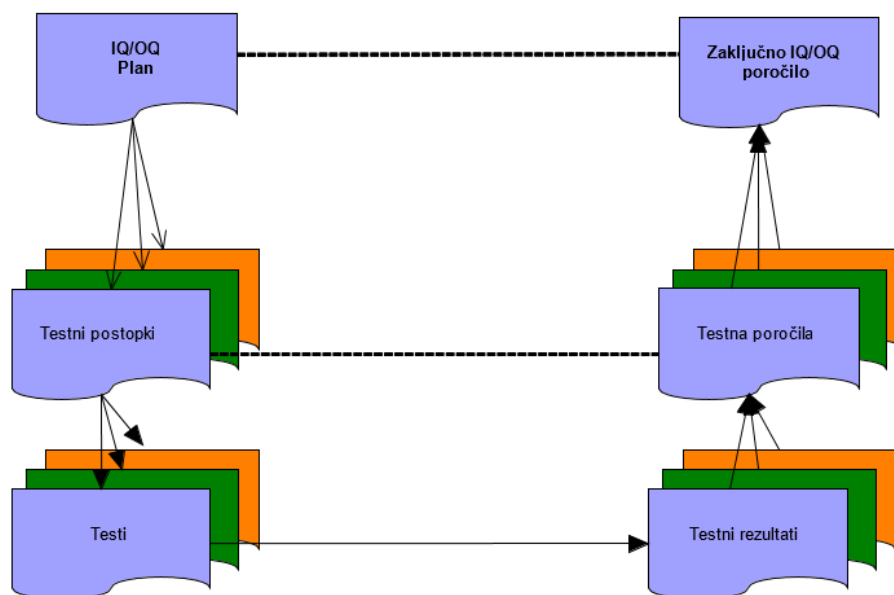
- uporabniške skupine in njihov nivo dostopa, kar je zelo pomembno pri potrjevanju in podpisovanju receptov ter generiranih poročil,
- konfiguracijo minimalnih in maksimalnih parametrov za vsako izmed naprav, kar je ključnega pomena pri omejevanju uporabnikovega vnosa parametrov in izogibanju napakam.

6.6 Testiranje

Testiranje je zadnji korak pri izvedbi novega projekta. Testirati je potrebno vse implementirane in spremenjene funkcionalnosti sistema. Testiranje izvajamo s pomočjo posameznih testov, pri čemer vsak izmed njih pokriva določeno področje. Testiranje ločimo na dva dela:

- kvalifikacijo namestitve (angl. Installation Qualification, IQ),
- kvalifikacijo obratovanja (angl. Operational Qualification, OQ).

Vsako izmed testiranj pa je sestavljeno iz treh različnih tipov dokumentov, razvidno iz slike 6.2:



Slika 6.2: Prikaz testnih dokumentov in njihov potek.

- plana testiranja oziroma kvalifikacije, ki opisuje osnovni koncept in definira testne postopke,
- testnih postopkov, namenjenih izvedbi specifičnih testov,
- zaključnega poročila testiranja, ki je skupen dokument vseh testov in podpisnikov ter vsebuje tabelo vseh uspešno zaključenih testov.

Testiranje je uspešno opravljeno, ko je izpolnjena celotna tabela testov. Vsak izmed testnih postopkov pa je sestavljen iz naslednjih delov: namena testiranja, testnih pogojev, testnih procedur, kriterija sprejemljivosti, testne tabele, liste odstopov in prilog. Pred izvedbo testiranja nam morajo podpisati vse testne dokumente. S tem nam nadzor kvalitete v posameznih dužbah potrди pokritost posameznih dodanih ali spremenjenih funkcionalnosti sistema.

6.6.1 Kvalifikacija namestitve

Kvalifikacija namestitve (angl. Installation Qualification, IQ), preverja ustreznost nameščenih delov sistema. Tu se preverjajo inštalacije, konfiguracije, zaslonske slike in podobno. V našem primeru smo predvideli štiri testne postopke.

1. Pri testu namestitve preverimo nastavitve spletnega strežnika, aplikacije za upravljanje z recepti. Preverimo nastavitve mrežnih kartic in povezav v omrežja, testiranje povezave na strežnik Historian in strežnik Microsoft SQL. Preverimo namestitev strežnika Kepware OPC in LinkMaster, preverimo zagon opravila za prenašanje poročil FTP in podobno.
2. Pri testu zaslonskih slik preverimo ustreznost povezav med vnosnimi obrazci, okno za nalaganje receptov, prikaz zgodovine dogodkov in poročil, ki so del aplikacije za upravljanje z recepti. S tem se preverimo tudi ustreznost tabel SQL in vnosnih polj.
3. S testom konfiguracije zaščite preverimo konfiguracije in delovanje sistema zaščite na aplikacijskem nivoju. V sklopu testiranja preverimo tudi zgodovino dogodkov kreiranih in spremenjenih receptov ter prijavo domenskih uporabnikov ob padcu domene. Prav tako testiramo tudi dostop do spletnega strežnika aplikacije za upravljanje z recepti.
4. Test komunikacije in konfiguracije naprav izvedemo ob vsakem priklopu novega stroja v sistem za upravljanje z recepti. Preverimo nastavitve mrežne povezave, nastavitve strežnika KEPServerEX OPC in LinkMaster ter ustreznost nameščene programske opreme, vključno z nastavitvami dostopa DCOM.

6.6.2 Kvalifikacija obratovanja

Kvalifikacija obratovanja (angl. Operational Qualification, OQ) je namenjena testiranju delovanja samega sistema. Testirajo se njegove posamezne

funkcionalnosti, kot so e-podpis, izpad opreme, obnovitev sistema in podobno. Izvedli smo naslednje štiri testne postopke.

1. S testom e-zapisov in e-podpisov preverimo zahteve za e-podpise in e-zapise iz uporabniških zahtev, ki se nanašajo na dokument FDA (angl. Food and Drug Administration) uprave ZDA za hrano in zdravila 21CFRPart11.
2. S testom izpada opreme preverimo degradacije sistema ob izpadih in nenormalnih dogodkih ter obstoj zasilnih režimov in mehanizmov, ki omogočajo zasilno delovanje sistema kljub izpadu. Testiramo tudi izklop in ponovni vklop sistema v realnem okolju.
3. S testom obnovitve sistema preverimo kreiranje slike diska (angl. disk image) ter obnovitev sistema iz kreirane slike na novih diskih. Testiramo delovanje komunikacije med sistemom za upravljanje z recepti in strojem.
4. Test funkcionalnosti stroja izvedemo ob vsakem priklopu novega stroja v sistem za upravljanje z recepti. Preverimo možnost nalaganja receptov (uspešno, neuspešno nalaganje), avtomatsko hranjenje kronologije dogodkov in procesnih podatkov stroja, avtomatski prenos poročil.

6.7 Uporabniška navodila

Poleg vseh dokumentov, ki opisujejo sistem, moramo pripraviti tudi navodila, ki so sestavljena iz dveh dokumentov.

- S sledenjem namestitvenim navodilom mora biti ustrezno usposobljena oseba zmožna namestiti vso programsko opremo, potrebno za delovanje sistema. Tu je vključena tudi konfiguracija, ki jo opravimo med namestitvijo.

- Navodila za uporabo morajo biti dovolj jasna, da lahko usposobljen uporabnik skonfigurira nameščen sistem. V našem primeru je v njih opisan postopek dodajanja novega stroja v obstoječi sistem.

6.8 Način obvladovanja dokumentov

Vse dokumente povežemo z dokumentom matrika sledljivosti (angl. Traceability Matrix, TM). Sestavljena je iz preproste tabele, ki vsebuje reference na:

- uporabniške zahteve v katerih je številka in kratek opis zahteve,
- funkcijska specifikacija, ki vsebuje dokument in poglavje, v katerem je opisano delovanje posamezne funkcionalnosti glede na točko iz uporabniških zahtev,
- specifikacijo programske ali strojne opreme, ki vsebuje dokument in poglavje, kjer je podrobneje opisano delovanje glede na zahtevo iz uporabniških zahtev,
- testne postopke/teste v katerih je testirana zahteva iz uporabniških zahtev.

V matriki sledljivosti morajo biti pokrite vse točke iz uporabniških zahtev, ki morajo biti popisane v zgoraj omenjenih dokumentih, poleg tega pa morajo biti tudi vse stestirane. Vedno moramo podati dokument tako, da sta iz reference jasno razvidna njegovo ime in verzija.

Prav tako imamo po priporočilih GAMP5 določena pravila pri izdelavi in izpolnjevanju projektne dokumentacije:

- na vsakem listu podamo trenutno stran in število vseh strani,
- za izpolnjevanje projektne dokumentacije obvezno uporabljamo neizbrisljiv kemični svinčnik, ki je priporočljivo modre barve,

- pri vsakem podpisu dodamo tudi datum, ki mora biti v formatu dd.mm.llll,
- napake pri izpoljevanju jasno prečrtamo in jih podpišemo, poleg obvezno dodamo tudi datum in, če imamo prostor, vpišemo še komentar z razlogom za nastalo napako,
- polja, ki jih pustimo prazna, prečrtamo,
- polja, ki nimajo vrednosti, označimo z ni aplicirano, N.A.

Poglavje 7

Sklepne ugotovitve

S postavitvijo enotnega sistema za upravljanje z recepti smo pospešili celoten postopek manipulacije in nalaganja receptov. Zdaj je centraliziran in zahvaljujoč spletni aplikaciji, enostavno dostopen. Poleg tega lahko med izdelovanjem enega proizvoda izpeljemo celoten postopek kreiranja in nalaganja recepta za nek drug proizvod. Arhitektura sistema uporablja koncept odjemalec-strežnik, kar nam omogoča enostavno dodajanje novih strojev.

Spletna aplikacija temelji na podatkovni bazi Microsoft SQL in nam omogoča popolno manipulacijo z recepti. To pomeni, da lahko v njej podpisujemo in potrjujemo recepte, jih kreiramo in jim nastavljamo parametre ter jih nalagamo na stroje. Implementiran je celoten postopek nalaganja in shranjevanja receptov. Poleg tega nam omogoča sledenje spremembam in njihov pregled.

Ciljna arhitektura obsega strežnika OPC DA in FTP na strojih, komunikacijski strežnik OPC in strežnik Microsoft SQL s podatkovno bazo na komunikacijskem računalniku. Poleg omenjenih strežnikov moramo zaradi zahtev po integriteti podatkov vključiti še dva strežnika. To sta strežnika Proficy Historian DA in Proficy Historian A&E, pri čemer prvi služi zajemu procesnih podatkov, drugi pa zajemu alarmov in dogodkov. Vsak stroj mora imeti nameščena tudi zbiralnika za oba strežnika.

Podatke o proizvodnih receptih na stroje naložimo iz podatkovne baze Mi-

Microsoft SQL. Naložimo jih preko komunikacijskega strežnika OPC na strežnik OPC DA, ki je nameščen na strojih. Poleg tega se po izvedenem receptu na strojih generira poročilo v obliki datoteke pdf, ki ga preko protokola FTP prečrpamo na komunikacijski računalnik. To nam omogoča, da lahko do njega dostopamo iz spletne aplikacije, v kateri ga lahko podpišemo in natisnemo.

Celoten sistem za upravljanje z recepti po standardu ISA 95 hierahično spada na informacijski nivo 3 in je dokumentiran po dokumentacijskih priporočilih GAMP5.

Aplikacijo zaradi njene zasnove lahko enostavno nadgradimo. Dodamo ji lahko funkcionalnosti, kot so: grafični prikaz podatkov in analitske funkcije. Lahko pa jo razširimo na nivoju strojev in povezav. Dodamo ji lahko nove gonilnike za stroje, ki ne podpirajo protokola OPC DA.

Literatura

- [1] IIS Dosegljivo
https://en.wikipedia.org/wiki/Internet_Information_Services [Dostopano 26.08.2016]
- [2] KEPServerEX, dosegljivo
<https://www.kepware.com/products/kepserverex/> [Dostopano 26.08.2016]
- [3] Kepware LinkMaster, dosegljivo
<https://www.kepware.com/products/linkmaster/> [Dostopano 26.08.2016]
- [4] OPC DA, dosegljivo
<https://opcfoundation.org/developer-tools/specifications-classic/data-access> [Dostopano 04.04.2016]
- [5] OPC UA, dosegljivo
<https://opcfoundation.org/developer-tools/specifications-unified-architecture> [Dostopano 04.04.2016]
- [6] ISPE, GAMP 5 A Risk-Based Approach to Compliant GxP Computerized Systems, 2008, Appendix M3 Science Based Quality Risk Management str. 105-127
- [7] ISPE, GAMP A Risk-Based Approach to Testing of GxP Systems, Second Edition 2012, Appendix T3 - Test Specifications, Cases, and Scripts str. 91-103

- [8] Proficy Historian, dosegljivo:
<http://www.geautomation.com/download/proficy-historian-datasheet>
[Dostopano 26.08.2016].
- [9] Proficy Historian DA, dosegljivo:
http://help.geautomation.com/Historian55/iHistorian.htm#../Subsystems/iHistCollMaster/Subsystems/iHISTOPCOL/content/dc_opc_data_collectors.htm%3FTocPath%3DHistorian%2520Data%2520Collectors%7COPC%2520Data%2520Collectors%7C____0
[Dostopano 26.08.2016].
- [10] Proficy Historian A&E, dosegljivo:
http://help.geautomation.com/Historian55/iHistorian.htm#../Subsystems/iHistAE/content/ihistorian_alarms_and_events.htm%3FTocPath%3DHistorian%2520Alarms%2520and%2520Events%7C____0 [Dostopano 26.08.2016].
- [11] ISA 95, dosegljivo:
<http://image.slidesharecdn.com/manufacturingoperationmanagement-150505053557-conversion-gate02/95/manufacturing-operation-management-25-638.jpg?cb=1430804274> [Dostopano 27.08.2016].
- [12] ISA 95, dosegljivo:
<https://en.wikipedia.org/wiki/ANSI/ISA-95> [Dostopano 27.08.2016].